UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,199	10/23/2003	Todd R. Manion	306341.01	8428

69316 7590 08/26/2010
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052

EXAMINER	
RASHID, HARUNUR	

ART UNIT	PAPER NUMBER
2458	

NOTIFICATION DATE	DELIVERY MODE
08/26/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

DBOUTON@MICROSOFT.COM
vffiling@microsoft.com
stevensp@microsoft.com

Office Action Summary	**Application No.** 10/692,199	**Applicant(s)** MANION ET AL.	
	Examiner HARUNUR RASHID	**Art Unit** 2458	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 October 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 October 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *8/11/08, 6/19/06, 6/6/05, 1/2/04*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-23 are pending in this examination.

Allowable Subject Matter

2. Claim 19 would be allowable if rewritten in independent form including all of the limitations of the claims 20, 21, 22, and 23.

Information Disclosure Statement

3. The IDS dated 8/11/08, 6/19/06, 6/6/05 and 1/2/04 has been considered. See PTO-1449.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-18 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

4. Referring to claims 1, 6, 13 and 16 the claim recites, “PeerName managed class comprising a peer name string property field, an authority field, a classifier field, and a secured field …”. “PeerName managed class” just limited to software modules or block

per se, as evidenced by the Specification ([0067]). The claim is not limited to statutory subject mater and is therefore nonstatutory.

5. Likewise, claims 2-5, 7-12, 14-15, 17 and 18 are dependent claims that depend on claims 1, 6, 13 and 16 and fail to resolve the above problems, therefore, claims 2-5, 7-12,14-15, 17 and 18 are also rejected under 35 U.S.C 101.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. The specification is objected to under 35 U.S.C. 112, first paragraph, as failing to adequately teach how to make an/or use the invention. The specification is enabling for a portion of the subject matter claimed but the enablement is not commensurate in scope with the claim. Specifically, the specification fails to show how the single step of "managed class…" of the claim can perform the claimed functions. Thus, it would require undue experimentation for a person having ordinary skill in the pertinent art to make and use the invention as disclosed and claimed.

Claims 1-18 are rejected under 35 U.S.C. 112 first paragraph, for the reasons set forth in the objection to the specification. See In re Hyatt 218 USPQ 195 (CAFC 1983).

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 1-18 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim language is very unclear and indefinite to understand the claimed subject matter.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pabla et al. (herein after Pabla) U.S. PGPub No.: 20040064693, in view of Joseph Mayo, C# Unleashed, Volume 2002, Pub. Date: November 14, 2001, ISBN-10:0-672-32122-X.

8. Referring to claim 1, Pabla discloses a PeerName managed class comprising a peer name string property field, an authority field, a classifier field, and a secured field (figs. 9-11, [0015], user name, peer identifier and/or name, a password, certificate, and

other authentication information and usage information. Identity information such as user names, peer identifiers and/or names, passwords, certificates and associated information may be stored in a distributed index, also see [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Referring to claim 2, Pabla discloses the PeerName managed class of claim 1, further comprising a first PeerName constructor to allow an application to specify the entire PeerName [0335], constructor, also see [0015] , [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 3, Pabla discloses the PeerName managed class of claim 1, further comprising a second PeerName constructor to allow an application to specify a PeerName based on peer identity information and a classifier to allow resolution of the PeerName to a third party [0015], peer identifier, also see [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 4, Pabla discloses the PeerName managed class of claim 1, further comprising a third PeerName constructor to allow an application to specify a PeerName based on a PeerName object and a classifier to allow resolution of the PeerName to a third party ([0016], peer group membership, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 5, Pabla discloses the PeerName managed class of claim 1, further comprising at least one method exposed thereby selected from the group consisting of an equals method, ==method, a GetHashCode method, a GetType method, a ReferenceEquals method, and a ToString method ([0015], hash, also see [0070], [0085], [0087], [0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 6, Pabla discloses a PeerIdentity managed class comprising a PeerName field, a FriendlyName field, and a Key field ([0015], user name, peer identifier and/or name, a password, certificate, and other authentication information and usage information. Identity information such as user names, peer identifiers and/or names, passwords, certificates and associated information may be stored in a distributed index, also see [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Referring to claim 7, Pabla discloses the PeerIdentity managed class of claim 6, further comprising a first constructor to retrieve an identity based on a PeerName parameter ([0016], find corresponding identity information, also see[0015] , [0070],[0085], [0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 8, Pabla discloses the PeerIdentity managed class of claim 6, further comprising a second constructor to create a new identity based on a friendlyName parameter, a classifier, and a key ([0015], the user identity may be added to the distributed index by adding identity information to the distributed index at the location of the distributed index corresponding to the hash of the user identity, also see [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 9, Pabla discloses the PeerIdentity managed class of claim 6, further comprising a third constructor to create a new identity based on friendlyName parameter and a classifier ([0352] create modules with a similar functionality, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 10, Pabla discloses the PeerIdentity managed class of claim 6, further comprising a fourth constructor to create a new identity based on a friendlyName parameter, a blank classifier, and a new key pair ([0352] create modules with a similar functionality, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Claim 11, is rejected for similar reasons as stated above.

Referring to claim 12, Pabla discloses the PeerIdentity managed class of claim 6, further comprising at least one static method exposed thereby selected from the group consisting of a GetIdentities static method, a first import static method utilizing an ExportedPeerIdentity and a password to import a PeerIdentity, and a second import static method utilizing an ExportedIdentityXml string and a password to import a PeerIdentity ([0127], multi-platform, secure execution environment, also see [0015],[0070],[0085],[0087], [0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 13, Pabla discloses a PeerIdentityInfo managed class comprising a PeerName field, a FriendlyName field, and a Key field ([0015], the user identity may be added to the distributed index by adding identity information to the distributed index at the location of the distributed index corresponding to the hash of the user identity, also see [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Referring to claim 14, Pabla discloses the PeerIdentityInfo managed class of claim 13, further comprising a PeerIdentityInfo constructor that constructs a PeerIdentityInfo object from an IdentityInfoXml string ([0352] create modules with a similar functionality, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Claim 15, is rejected for similar reasons as stated above.

Referring to claim 16, Pabla discloses an ExportedPeerIdentity managed class comprising an ExportedPeerIdentity constructor that utilizes an ExportedIdentityXmlString to construct an ExportedPeerIdentity object (figs. 9-11, [0015], user name, peer identifier and/or name, a password, certificate, and other authentication information and usage information. Identity information such as user names, peer identifiers and/or names, passwords, certificates and associated information may be stored in a distributed index, also see [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Referring to claim 17, Pabla discloses the ExportedPeerIdentity managed class of claim 16, further comprising a PeerName property that returns the PeerName of the exported identity ([0352] create modules with a similar functionality, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Claim 18, is rejected for similar reasons as stated above.

Referring to claim 19, Pabla discloses a method of managing by an application a PeerName in a managed framework, the method comprising the steps of: communicating with a managed PeerName object, the managed PeerName object exposing constructors for creating an entire PeerName object, creating a hypothesized PeerName object for resolution to a third party from PeerIdentityInfo and a classifier, and creating a second hypothesized PeerName object for resolution from a known PeerName object; selecting one of the constructors; passing to the managed PeerName object parameters required by the constructor selected; and initiating the constructor ([0065] A first peer may initially handle all the space in the distributed index. When another peer joins the distributed index, the distributed index may be split between the two peers to create two spaces or zones of the distributed index, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Referring to claim 20, Pabla discloses the method of managing by an application a PeerIdentity in a managed framework, the method comprising the steps of: communicating with a managed PeerIdentity object, the managed PeerIdentity object exposing constructors for creating a new identity from parameters of FriendlyName, classifier, and key, for creating a new identity from parameters of FriendlyName and classifier, for creating a new identity from parameters of FriendlyName, a blank classifier, and a new key pair, and retrieving an identity associated with a given PeerName parameter; selecting one of the constructors; passing to the managed PeerIdentity object parameters required by the constructor selected; and initiating the constructor ([0352] create modules with a similar functionality, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Referring to claim 21, Pabla discloses the method of claim 20, wherein the managed PeerIdentity object further exposes static methods for importing a peer identity from an ExportedPeerIdentity and a password, for importing a peer identity from an ExportedIdentityXml string and a password, and for retrieving a collection of identities for a user account, the method further comprising the steps of selecting one of the static methods, passing to the managed PeerIdentity object parameters required by the static method, and initiating the static method (figs. 9-11, [0015], user name, peer identifier and/or name, a password, certificate, and other authentication information and usage information. Identity information such as user names, peer identifiers and/or names, passwords, certificates and associated information may be stored in a distributed index, also see [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Referring to claim 22, Pabla discloses a method of managing by an application PeerIdentityInfo in a managed framework, the method comprising the steps of: communicating with a managed PeerIdentityInfo object, the managed PeerIdentityInfo object exposing a constructor for creating a PeerIdentityInfo object from an IdentityInfoXml parameter; passing to the managed PeerIdentityInfo object the parameter required by the constructor; and initiating the constructor ([0352]

create modules with a similar functionality, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Referring to claim 23, Pabla discloses a method of managing by an application a ExportedPeerIdentity in a managed framework, the method comprising the steps of: communicating with a managed ExportedPeerIdentity object, the managed ExportedPeerIdentity object exposing a constructor for creating an ExportedPeerIdentity object from an ExportedIdentityXmlString; passing to the managed ExportedPeerIdentity object parameters required by the constructor; and initiating the constructor ([0352] create modules with a similar functionality, also see [0015], [0070],[0085],[0087],[0098-0099],[0323],[0332-0333],[0469-0471],[0617]).

Pabla does not explicitly disclose class as a code format. However, Mayo discloses class as a code format (pages 86 and 124).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teaching of Pabla with the teaching of Mayo by including the feature of class as a code format, in order for Pabla's system to able to use programming codes to manage peer-to-peer network as a result all computers share responsibility, which may include bandwidth, storage space, and computing power and avoid central database thereby increasing user desirability of use.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

10. Examiner's Note: Examiner has cited particular columns/paragraphs/pages and line numbers in the references as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in its entirety as potentially teaching of all or part of the claimed invention.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to HARUNUR RASHID whose telephone number is

(571)270-7195. The examiner can normally be reached on Monday - Friday 9:00 AM to 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph E. Avellino can be reached on 571-272-3905. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/H. R./
Examiner, Art Unit 2458

/Joseph E. Avellino/
Supervisory Patent Examiner, Art Unit 2458